Vol. II, No. 4
March/April, 1983

SATN®

A Publication of
Software Arts Products Corp.

Software Arts Technical Notes from the creators of VisiCalc®

A Word About Our New Look

While designing TK!SATN, Software Arts' new publication for users of the TK!Solver program, we decided to provide readers with a "map" to each issue. The cover we adopted is like this one, which replaces the *In This Issue* box and *About This Issue* article that led off each issue of SATN. Each article in each issue of both SATN and TK!SATN will be described briefly and each description will be accompanied by a tabular listing of the functions and commands that the article uses. We hope that you will find this cover more useful than its predecessor and would like to hear your comments.

CONTENTS	FEATURING	PAGE
Turning the Tables on Taxes This article looks at the problem of using the VisiCalc program to choose one of several tables. Given the time of year, the article concentrates on Federal tax rate schedules. One of the goals we set for ourselves when we developed the article was to use functions available in all versions of the program.	@INT @LOOKUP @MAX @SUM /R /S#L /S#S	2
Book Review Richard Foard, who also reviews science books for the Baltimore Sun, reviews *VisiCalc for the IBM Computer*, by Edouard J. Desautels, one of a series of books about using the VisiCalc program by the same author.		7
VisiCalc and WordStar Craig Stinson continues his series on using the VisiCalc program with word processing programs. In this issue he looks at the use of WordStar for the IBM.	/PF	8
Related Products Checklist Once again we list some of the products that enhance the VisiCalc program.		10
VisiCalc Advanced Version We examine how some of the features of the VisiCalc Advanced Version program may be used when developing the sheet used in *Turning the Tables on Taxes.* Though the expressions on both sheets are identical, using Advanced Version allows us to display only the information we choose to display. Using variable column width also permits us to view the entire sheet, and so carry out a series of replications quite easily.	/A /GA /GCC /K (^K=)	12

TURNING THE TABLES ON TAXES

Dealing with tax rate schedules for individual income tax returns is one of the less tractable problems VisiCalc users confront. Estates and trusts use a single schedule; corporations are taxed according to a formula. But four individuals, all having the same income, might use four different schedules to calculate their taxes. Those schedules, shown in Table 1, include most of the same numbers for income, base taxes, and marginal tax rates. Because of such overlapping, many users finally settle on the less than satisfactory expedient of entering each schedule in a separate file and, depending on the status of the taxpayer, loading the file that applies.

One difficulty of dealing with the schedules, and other tables like them, is that we have grown too accustomed to their forms. Such tables are most easily used when we have rulers in our hands. We only rarely think to rewrite them, and often wind up forcing the VisiCalc program to accommodate the tables rather than rewriting the tables so that they can be used efficiently by the program. The purpose of this article is to show you one way to rewrite tables so that they can be consolidated for use within the VisiCalc program.

Although the major example used in this article is the Federal income tax rate schedules for individuals, the essential idea is quite simple and can be used in any other application in which you must manipulate several related tables containing overlapping values.

Rewriting the Schedules

Our aim is to create a single, three-line table that we can use with @LOOKUP to find the base tax and marginal tax rate for a given filing status and taxable income amount.

Start at C1 by entering the values shown in color in Table 1. Referring to Figure 1, begin by entering on row 1 the values for married individuals filing separate returns; on row 2, the values for unmarried individuals; on row 3, for heads of households; and on row 4, married individuals filing joint returns. We chose the order of the filing statuses because it produced a roughly ascending series of values. They can be reordered, however, if that suits your purposes.

Next, because the tax rate schedules define each bracket as "over X dollars, but not over Y dollars," and the Y dollar amount defines the beginning of the next bracket, you must add .01 to the values just entered. So, at C6 enter and replicate:

```
>C6:.01+C1®/R:C7...C9:R
```

Now replicate that sequence across the sheet. Enter:

```
>C6:/R...C9:D6...06:RRRR
```

That will produce .01 at 06, 08, and 09, which you should blank with /B.

Beginning at C11, make the next series of entries from the shaded sections of the rate schedules shown in Table 1, under the heading @LOOKUP VALUE. Those values do not appear in any of the schedules produced by the government. However, except for the value 12.12, which begins each sequence, the @LOOKUP VALUE that we have constructed is derived from the schedules. Because each base tax amount is uniquely associated with a rate, we may define a component value by adding the base tax, an integer, and the tax rate, a decimal. For example, a rate of 14% and a tax of $126 yield 126.14. That value, of course, is neither a dollar amount nor percentage, but by using @INT can be restored quite simply to its two component parts.

Key to Conventions Used in SATN

The VisiCalc® program is available on many different personal computers. The keyboards on the machines vary slightly. To reduce confusion, SATN will use the symbol ® and the word RETURN synonymously to represent the keys known as ENTER, ENDLINE, and RETURN. The symbol ^ is used to represent the exponentiation operator.

The VisiCalc program was designed so that the colon (:) has the same effect as RETURN in some commands, as when ending the > command and to separate the various elements of the replicate command. SATN uses the same convention.

In examples of what to type, type exactly what you see and press RETURN at the end of every line. On some computers, you may find it necessary to add parentheses to some of the expressions included in the text. For example, 3+4*5 may become (3+4)*5.

Throughout SATN, one typeface represents both VisiCalc input and output (what you should type and what you will see on the screen). The typeface looks like this:

```
This line is typeset in OCR-B.
So is this: +A1+B1*3.25
```

To allow you to type the formulas directly into VisiCalc, the examples use specific coordinates, like A1, rather than the names of the value they represent. When actually typing a formula, you can use the arrows to point rather than using the > command.

SATN is published bimonthly by Software Arts Products Corp., 27 Mica Lane, Wellesley, MA 02181. Address all editorial correspondence to the Editorial Department, SATN, 27 Mica Lane, Wellesley, MA 02181. Requests for special permission or bulk orders should be addressed to the publisher.

Domestic subscriptions: 6 issues $30.00, 12 issues $50.00; outside U.S.A., 6 issues $50.00, 12 issues $90.00. Single copy price, $6.00. Send subscriptions, change of address, and fulfillment questions to SATN Subscriptions, P.O. Box 815, Quincy, MA 02169.

Copyright © 1983, Software Arts Products Corp. All rights reserved. This periodical may not be copied or reproduced in any fashion either in whole or in part without the prior written consent of Software Arts Products Corp.

Software Arts Products Corp. makes no warranty, express or implied, with respect to the quality, accuracy or freedom from error of the information contained in SATN, the journal for VisiCalc users, and specifically disclaims all liability for any damages resulting from the use of such information.

Unless specifically stated otherwise in any article in this issue of SATN, Software Arts Products Corp. has not examined any products and/or publications identified in SATN to determine their accuracy and/or ability to perform as advertised. Mention of a product and/or publication in SATN does not constitute an endorsement by Software Arts Products Corp. of any such product or publication.

Software Arts is a trademark of Software Arts, Inc. and Software Arts Products Corp. DIF is a trademark of Software Arts Products Corp. SATN is a registered trademark of Software Arts Products Corp. VisiCalc is a registered trademark of VisiCorp.

Apple is a registered trademark of Apple Computer, Inc. IBM is a registered trademark of International Business Machines, Inc. Atari is a trademark of Atari, Inc. Commodore is a trademark of Commodore International. HP-83 and HP-85 are trademarks of Hewlett-Packard Company. TRS-80 is a trademark of Radio Shack division of Tandy Corp.

Keystroke Memory is a trademark of Software Arts Products Corp.

Status 1 Married, Filing Separately

Taxable Income				
Over	Not Over	Base Tax	Marginal Rate	@LOOKUP VALUE
0.00	1,700	0	0	—
1,700	2,750	0	12%	12.12
2,750	3,800	126	14%	126.14
3,800	5,950	273	16%	273.16
5,950	8,000	617	19%	617.19
8,000	10,100	1,006	22%	1,006.22
10,100	12,300	1,468	25%	1,468.25
12,300	14,950	2,018	29%	2,018.29
14,950	17,600	2,787	33%	2,787.33
17,600	22,900	3,661	39%	3,661.39
22,900	30,000	5,728	44%	5,728.44
30,000	42,800	8,852	49%	8,852.49
42,800	—	15,124	50%	15,124.50

Status 2 Unmarried

Taxable Income				
Over	Not Over	Base Tax	Marginal Rate	@LOOKUP VALUE
0.00	2,300	0	0	—
2,300	3,400	0	12%	12.12
3,400	4,400	132	14%	132.14
4,400	6,500	272	16%	272.16
6,500	8,500	608	17%	608.17
8,500	10,800	948	19%	948.19
10,800	12,900	1,385	22%	1,385.22
12,900	15,000	1,847	23%	1,847.23
15,000	18,200	2,330	27%	2,330.27
18,200	23,500	3,194	31%	3,194.31
23,500	28,800	4,837	35%	4,837.35
28,800	34,100	6,692	40%	6,692.40
34,100	41,500	8,812	44%	8,812.44
41,500	—	12,068	50%	12.068.50

Status 3 Heads of Households

Taxable Income				
Over	Not Over	Base Tax	Marginal Rate	@LOOKUP VALUE
0.00	2,300	0	0	—
2,300	4,400	0	12%	12.12
4,400	6,500	252	14%	252.14
6,500	8,700	546	18%	546.18
8,700	11,800	898	20%	898.20
11,800	15,000	1,518	22%	1,518.22
15,000	18,200	2,222	23%	2,222.23
18,200	23,500	2,958	28%	2,958.28
23,500	28,800	4,442	32%	4,442.32
28,800	34,100	6,138	38%	6,138.38
34,100	44,700	8,152	41%	8,152.41
44,700	60,600	12,498	49%	12,498.49
60,600	—	20,289	50%	20,289.50

Status 4 Married, Filing Jointly

Taxable Income				
Over	Not Over	Base Tax	Marginal Rate	@LOOKUP VALUE
0.00	3,400	0	0	—
3,400	5,500	0	12%	12.12
5,500	7,600	252	14%	252.14
7,600	11,900	546	16%	546.16
11,900	16,000	1,234	19%	1,234.19
16,000	20,200	2,013	22%	2,013.22
20,200	24,600	2,937	25%	2,937.25
24,600	29,900	4,037	29%	4,037.29
29,900	35,200	5,574	33%	5,574.33
35,200	45,800	7,323	39%	7,323.39
45,800	60,000	11,457	44%	11,457.44
60,000	85,600	17,705	49%	17,705.49
85,600	—	30,249	50%	30,249.50

TABLE 1 *Although the order defined for entering filing statuses is roughly ascending, there's no requirement that this order be used. Enter the values that are printed in color, and in the shaded portion of the table.*

Creating such component values, especially if they can be disassembled so simply, makes it possible to construct relatively simple tables to be searched with @LOOKUP. The initial value in each sequence (12.12), however, is less clearly derived than the others. A small example can be used to explain both the source of the value and the general idea that we will use to rewrite the rate schedules.

Imagine that you want to create a single @LOOKUP table from these two sequences:

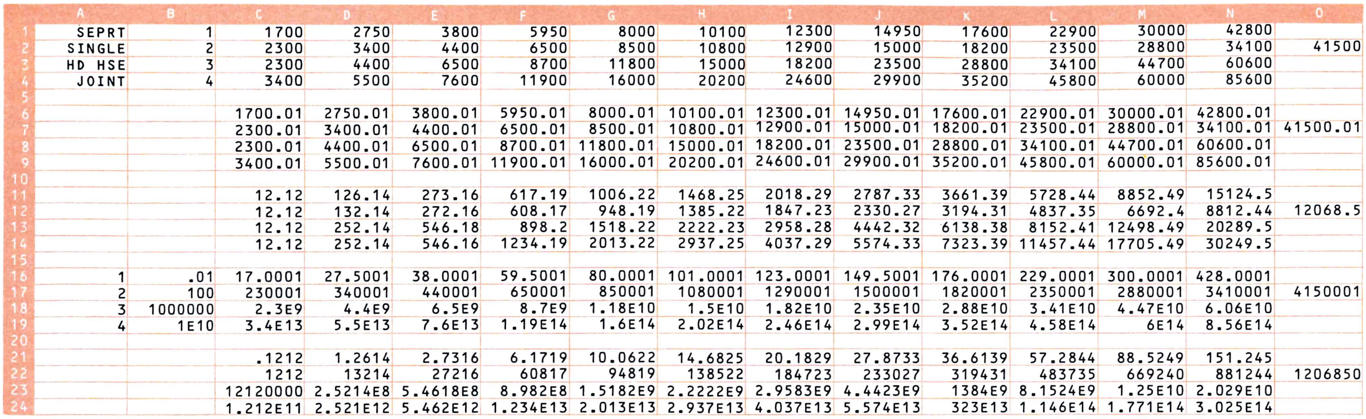
A		B	
1.10	11	1.26	.01
2.01	21	2.51	.03
3.01	8	3.74	.91
4.80	12	4.75	1.01
5.09	2	5.01	2.02

If you want to look up 1.99 in those tables, then you would get back 11 from Table A or .01 from Table B. Looking up 2.01 would return 21 or .01. The @LOOKUP function assumes that all the values in the range being searched are in ascending order, but you cannot merge Tables A and B and make the search range 1.10, 1.26, 2.01, . . . 5.01, 5.09, because you would not be able to look up intermediate values. For example, 1.99 would only bring back .01 and 2.01 would only bring back 21. Nor can you string the tables together so that the sequence 4.80, 5.09, 1.26, 2.51 appeared, because the @LOOKUP function stops at the first value greater than the one being looked up and then returns to the preceding value to select the corresponding value from the range to the immediate right of or immediately below the search range.

As shown in Figure 2, though, you can multiply the search range by factors that are large enough to insure that overlapping values will not occur in the range when you search for some value. In Figure 2, for example, Table A is multiplied by .01 and Table B by a factor that is 100,000 times larger; that is, by 1000. To look up some value, multiply it by the same factor used to multiply the table. The @LOOKUP function will find the appropriate value in the search range and will return the corresponding value in the range immediately to the right of or immediately below it. The actual expressions used are shown in the caption to Figure 2.

We have to massage the values we're searching for a bit before entering the range, but that's a minor consideration.

The tax rate schedules published by the IRS present a special problem. That is, the first bracket in



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SEPRT	1	1700	2750	3800	5950	8000	10100	12300	14950	17600	22900	30000	42800	
2	SINGLE	2	2300	3400	4400	6500	8500	10800	12900	15000	18200	23500	28800	34100	41500
3	HD HSE	3	2300	4400	6500	8700	11800	15000	18200	23500	28800	34100	44700	60600	
4	JOINT	4	3400	5500	7600	11900	16000	20200	24600	29900	35200	45800	60000	85600	
5															
6			1700.01	2750.01	3800.01	5950.01	8000.01	10100.01	12300.01	14950.01	17600.01	22900.01	30000.01	42800.01	
7			2300.01	3400.01	4400.01	6500.01	8500.01	10800.01	12900.01	15000.01	18200.01	23500.01	28800.01	34100.01	41500.01
8			2300.01	4400.01	6500.01	8700.01	11800.01	15000.01	18200.01	23500.01	28800.01	34100.01	44700.01	60600.01	
9			3400.01	5500.01	7600.01	11900.01	16000.01	20200.01	24600.01	29900.01	35200.01	45800.01	60000.01	85600.01	
10															
11			12.12	126.14	273.16	617.19	1006.22	1468.25	2018.29	2787.33	3661.39	5728.44	8852.49	15124.5	
12			12.12	132.14	272.16	608.17	948.19	1385.22	1847.23	2330.27	3194.31	4837.35	6692.4	8812.44	12068.5
13			12.12	252.14	546.18	898.2	1518.22	2222.23	2958.28	4442.32	6138.38	8152.41	12498.49	20289.5	
14			12.12	252.14	546.16	1234.19	2013.22	2937.25	4037.29	5574.33	7323.39	11457.44	17705.49	30249.5	
15															
16	1	.01	17.0001	27.5001	38.0001	59.5001	80.0001	101.0001	123.0001	149.5001	176.0001	229.0001	300.0001	428.0001	
17	2	100	230001	340001	440001	650001	850001	1080001	1290001	1500001	1820001	2350001	2880001	3410001	4150001
18	3	1000000	2.3E9	4.4E9	6.5E9	8.7E9	1.18E10	1.5E10	1.82E10	2.35E10	2.88E10	3.41E10	4.47E10	6.06E10	
19	4	1E10	3.4E13	5.5E13	7.6E13	1.19E14	1.6E14	2.02E14	2.46E14	2.99E14	3.52E14	4.58E14	6E14	8.56E14	
20															
21			.1212	1.2614	2.7316	6.1719	10.0622	14.6825	20.1829	27.8733	36.6139	57.2844	88.5249	151.245	
22			1212	13214	27216	60817	94819	138522	184723	233027	319431	483735	669240	881244	1206850
23			12120000	2.5214E8	5.4618E8	8.982E8	1.5182E9	2.2222E9	2.9583E9	4.4423E9	1384E9	8.1524E9	1.25E10	2.029E10	
24			1.212E11	2.521E12	5.462E12	1.234E13	2.013E13	2.937E13	4.037E13	5.574E13	323E13	1.146E14	1.771E14	3.025E14	

FIGURE 1 *This figure shows only a portion of the sheet. The labels in A1 . . . A4 may be replicated into A6 and below to help identify the various sets of figures.*

each filing status consists only of a rate, and no base tax. Given that, to combine the tax rate schedules as we combined the two tables in Figure 2, and to keep the tables from overlapping, the first multiple would have to be around 169,080, since 20,289.50/.12 is 169,079.2. However, 20,289.50/12.12 is 1,674.051, meaning that we can use smaller factors to combine the schedules. Basically, all that we've done is define a base tax of $12.00 (12% of $100) for the lowest taxable income amount. Later, we shall have to account for that deviation from the schedule.

The first 14 rows of your sheet should be identical to the ones shown in Figure 1. Now enter the factors that we will use to multiply the differing schedules. Enter:

```
>B16:.01
>B17:10000*B16®/R:B18 . . . B19:R
```

That creates a series of factors each of which is 10,000 times larger than the one before. Now, to apply those multipliers to the tax schedules, enter and replicate:

```
>C16:+C6*B16®/R:C17 . . . C19:RR
>C16:/R . . . C19:D16 . . . O16:RNRNRNRN
```

That produces the taxable income portion of the table. To produce the base tax and marginal tax rate portion of the table, enter and replicate:

```
>C16:/R . . . C19:C21:RNRNRNRN
>C21:/R . . . C24:D21 . . . O21:RNRNRNRN
```

Now compare the values in rows 16 . . . 24. You'll see, for example, that the highest value on row 16 (428.001 at N16) is lower than the lowest value on row 17 (230001 at C17). One way to check that you have made the entries correctly is to subtract N16 from C17 and then replicate that expression relatively down the sheet. The only negative value you should get will be caused by subtracting a value from a blank cell. Rows 16 . . . 24 now consist of series that end with values lower than the values that begin the series on the next row. Since they are now in a consistent, ascending order, all that has to be done to turn them into a usable @LOOKUP table is to place them on a single row. (We happen to have chosen to enter the original tables horizontally. It would not be difficult to restructure them into vertical tables.)

To create a single table by moving our newly calculated values around the sheet would not be very easy if we did not take another step. To make such moves, we would have to use /R, but with as many calculations as those tables contain we would have to enter a large number of value references. We used a DIF file to turn all the calculations into data. When you save a DIF file, only the values are saved, not the expressions that produced them. By reloading the DIF file atop the sheet that is currently on the screen (after first saving the underlying file) you can write over the calculated values, and then edit the sheet with /R without having to worry about value references.

There's another advantage to using a DIF file. When you substitute the DIF file for the calculations that produced it, you reduce memory requirements and recalculation time. In this instance, the savings are not substantial, but keep this technique in mind for other applications you may develop.

First, save the sheet you have created, giving it some filename that suits your purposes. You will not need that file until the Congress changes the tax laws again. When that happens, though, you will only have to plug in the new numbers, and the sheet will reproduce the rest.

With the cursor at C16 enter:

```
>C16:/S#STAX,DIF
```

The program will prompt you with DATA SAVE: LOWER RIGHT. Enter:

```
O24
```

You'll then be prompted with DATA SAVE: R C OR RETURN. Save the file by rows. Enter:

```
R
```

Assuming that you have already saved the /SS file, enter:

```
>C6:/S#LTAX.DIF
```

When you are prompted for Row, Column, or Return, enter R. By replicating row 15 into rows 16 through 24, you will clear the sheet. You will have overwritten the values in C6 through O14 with the values that were on C16 . . . O24. Since those cells are still on the sheet and contain calculations, they will recalculate. They are no longer needed, though, and may be erased by replicating a blank area into that part of the sheet.

Next, we have to consolidate the values in rows 2 . . . 4, 7 . . . 9, and 12 . . . 14 into rows 1, 6, and 11. To do that we use the Replicate command to copy row into row. Notice that when replicating a range into another range in the same direction, you have only to specify the opening coordinate of the target range. Enter:

```
>C4:/R . . . N4:O3
>C3:/R . . . Z3:P2
>C2:/R . . . AN2:O1

>C9:/R . . . N9:O8
>C8:/R . . . Z8:P7
>C7:/R . . . AN7:O6

>C14:/R . . . N14:O13
>C13:/R . . . Z13:P12
>C12:/R . . . AN12:O11
```

Now move rows 6 and 11 into place with the entries:

```
>C6:/MC1
>C11:/MC2
```

Finally, either delete rows 4 through 13 or replicate a range of blank cells into that range. That will leave you with three rows, beginning with 17.0001 at C1 and ending with 85600 at AY3. A portion of this can be seen in Figure 3.

Next, in order to account for the $12.00 tax that we added to the minimum marginal rate (12.12), change the opening sequences of each series in row 3. Enter:

```
>C3:1800
>O3:2400
>AB3:2400
>AO3:3500
```

Row 1 is used as the first @LOOKUP table and once a value is returned from row 1, that value is then used later to find a value in row 3, thus turning row 2 into an @LOOKUP table. Adjust the labels in A1 . . . B3 as shown in Figure 3.

Because we eliminated the table of multipliers when editing the sheet, it must be restored. Enter:

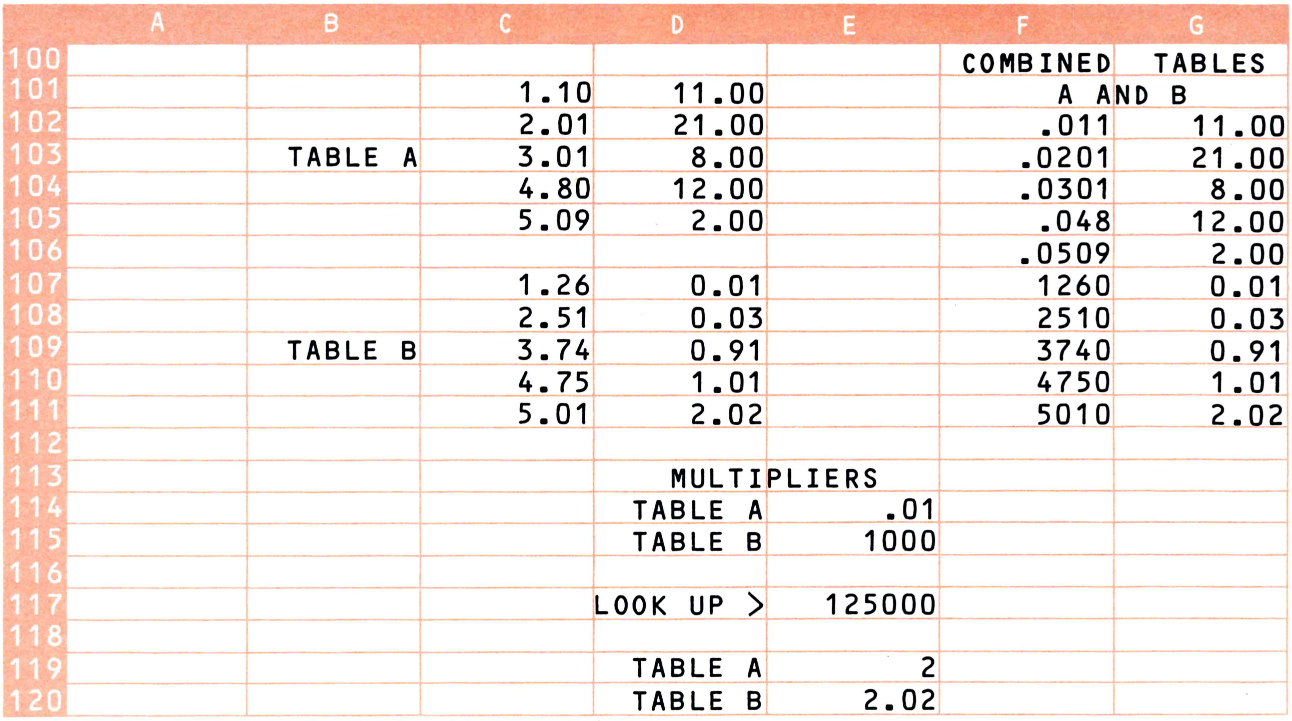
```
>C5:1
>C6:.01
>D5:1+C5
>D6:10000*C6
>D5:/R . . . D6:E5 . . . F5:RR
```

Notice that we have created a small @LOOKUP table consisting of the filing status and its associated multiplier.

All the values needed to calculate an individual's income tax are now on the sheet and in a form that allows us to quickly calculate and recalculate the tax due.

Calculating the Tax

The only two entered values that remain are the taxable income and the filing status. Although we have set up the sheet so that those values are entered at C14 and C15, you may already have developed a sheet in



	A	B	C	D	E	F	G
100						COMBINED	TABLES
101			1.10	11.00		A AND B	
102			2.01	21.00		.011	11.00
103		TABLE A	3.01	8.00		.0201	21.00
104			4.80	12.00		.0301	8.00
105			5.09	2.00		.048	12.00
106						.0509	2.00
107			1.26	0.01		1260	0.01
108			2.51	0.03		2510	0.03
109		TABLE B	3.74	0.91		3740	0.91
110			4.75	1.01		4750	1.01
111			5.01	2.02		5010	2.02
112							
113				MULTIPLIERS			
114				TABLE A	.01		
115				TABLE B	1000		
116							
117				LOOK UP >	125000		
118							
119				TABLE A	2		
120				TABLE B	2.02		

FIGURE 2 *The general expression:*

@LOOKUP(Value*Multiplier, Range)

is used here. If the elements that are being searched for, which appear in Column D, are also multiplied, then the expression would be

@LOOKUP(Value*Multiplier, Range)/Multiplier

The actual expression at E119 is @LOOKUP(E117*E114,F102 . . . F111), *which is replicated with the value references* NRNN *into E120.*

which the values have been calculated from a simulation of Form 1040. If so, you should adjust the value references accordingly.

The first calculation that must be made is to apply the appropriate multiplier to the taxable income. The second is to find the base tax and the marginal rate that are applied to the first calculation. One way to perform those operations would involve separate steps, but we have chosen to combine them into a single step, as it shows that you can nest one @LOOKUP statement within another and that you can look up an expression as well as a single value. The general expression is:

@LOOKUP(TAXABLE INCOME ∗
@LOOKUP(FILING STATUS))

which is entered at C16 as:

```
>C16:/F*@LOOKUP(C14*@LOOKUP(C15,
                      C5 . . . F5),C1 . . . AY1)
```

Notice as shown in Figure 3 that by formatting the cell with /F∗ the result appears to be primarily decorative, rather than functional.

The next step is to calculate the individual's marginal income. That uses the value that was returned by the expression at C16 and is simply the taxable income minus the excess tax amount. On our sheet it is entered as:

```
>C17:/F$+C14-@LOOKUP(C16,C2 . . . AY2)
```

Now, "hidden" from view behind another set of apparently decorative asterisks, is the expression used to prepare the @LOOKUP VALUE value for the base tax and marginal rate. That value was calculated with the compound @LOOKUP statement at C16 and must be reduced to the amount shown in Table 1 for the appropriate filing status. Since we originally constructed the @LOOKUP table by multiplying Table 1, we can retrieve the original values by dividing by the appropriate factor. Enter:

```
>C18:/F*+C16/@LOOKUP(C15,C5 . . . F5)
```

That returns the @LOOKUP VALUE, the Base Tax portion of which is calculated with @INT. Enter:

```
>C19:/F$@INT(C18)
```

Then, to get the Marginal Rate from the @LOOKUP VALUE is a simple subtraction. Since the rate is a percentage, we have multiplied it by 100 and entered a label in D20, identifying the value in C20 as a percentage. Enter:

```
>C20:+C18-C19*100
```

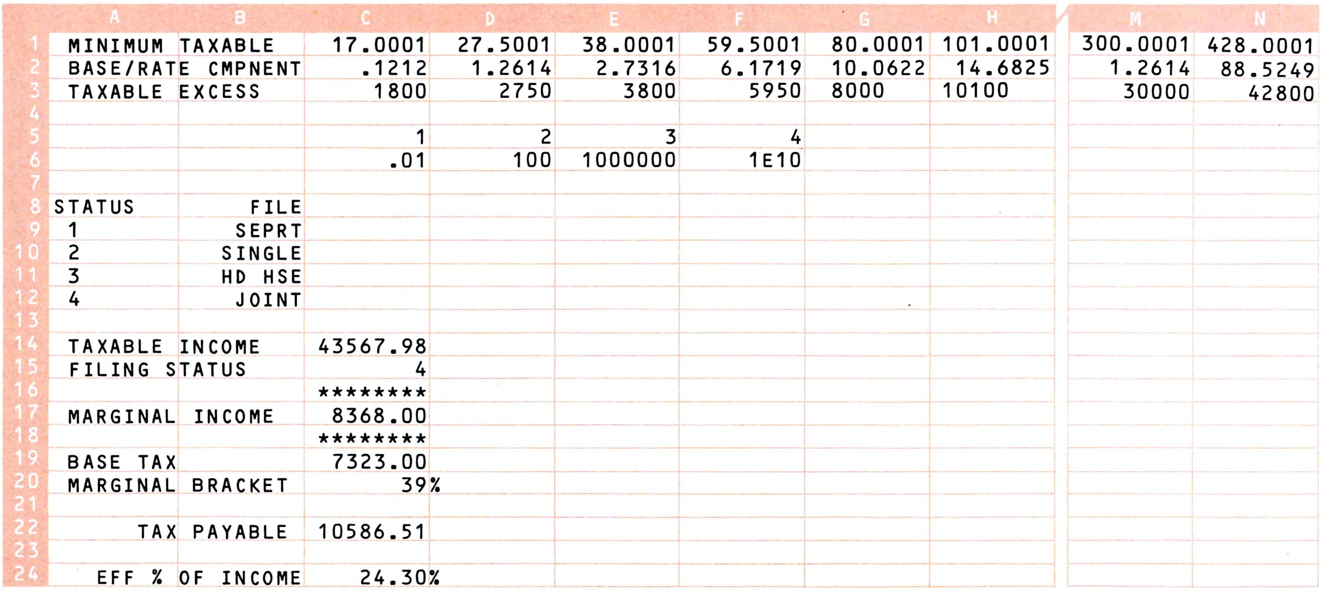
In general terms, the calculation of the tax is simply:

TAX PAYABLE = (MARGINAL INCOME ∗
MARGINAL BRACKET) + BASE TAX

As we have constructed the sheet, however, there are conditions under which the calculation of marginal income yields a negative value. That occurs when the amount entered at taxable income is less than the minimum amount for the particular filing status (except filing status 1, in which case NA is returned). If we were to use the general expression, shown above, then the sheet would show a negative tax, which could not be interpreted reasonably as money the Treasury owes the taxpayer. Since zero is larger than any negative number use @MAX to prevent calculating a negative tax. Enter:

```
>C22:/F$@MAX(C20/100*C17+C19,0)
```

Finally, notice that the multipliers we used are not so much larger than each other that it is impossible for an amount of taxable income to "slide" from, say, filing status 1 to filing status 2. For example, if you enter 25,000,000 at C14 and 1 at C15, marginal income will calculate to 24,997,600 because the multiplier for fil-



	A	B	C	D	E	F	G	H	M	N
1	MINIMUM	TAXABLE	17.0001	27.5001	38.0001	59.5001	80.0001	101.0001	300.0001	428.0001
2	BASE/RATE	CMPNENT	.1212	1.2614	2.7316	6.1719	10.0622	14.6825	1.2614	88.5249
3	TAXABLE	EXCESS	1800	2750	3800	5950	8000	10100	30000	42800
4										
5			1	2	3	4				
6			.01	100	1000000	1E10				
7										
8	STATUS	FILE								
9	1	SEPRT								
10	2	SINGLE								
11	3	HD HSE								
12	4	JOINT								
13										
14	TAXABLE	INCOME	43567.98							
15	FILING STATUS		4							
16			********							
17	MARGINAL	INCOME	8368.00							
18			********							
19	BASE TAX		7323.00							
20	MARGINAL	BRACKET	39%							
21										
22	TAX	PAYABLE	10586.51							
23										
24	EFF %	OF INCOME	24.30%							

FIGURE 3 *Although it is not covered in the text, the entry at C24, EFF % OF INCOME, is an interesting calculation in two ways. First, it highlights the difference between a taxpayer's marginal and actual tax brackets, and second, if you exceed the limits discussed in the text, it will show an unusually low value. That is, if the income should be entered as 40,000 and it is entered as 40,000,000 (in all statuses*

ing status 1 (.01) yields 250,000. That value, though, falls between 230,001 and 340,001, which is beyond the point at which the table for filing status 1 ends. Given the tax rate schedules and the multipliers the upper limits that can be used for each status are 23,000,099.99 for statuses 1 and 2, and 34,000,099.99 for status 3. The upper limit for status 4 is the upper limit of the program, or just under 1E62, which if translated into dollars would be more money than there is in the world.

If you inadvertently enter a taxable income that exceeds those limits, however, the marginal bracket will display 0%, because the component value returned by the expression, even after division by the appropriate multiplier, will still be an integer. Thus, the marginal bracket calculation also serves as an error trap.

We have used IRS tax rate schedules as examples because of the time of year. But for those of us who are not tax attorneys or accountants, we probably spend more time grumbling unhappily about taxes than we do using tax tables. Nonetheless, we often have other tables to work with: insurance premium or annuity tables, price lists, tables of densities and properties of materials, and so forth, all of which can be adapted to the technique we've outlined in this article. Keep in mind, too, that the program's limits of decimal precision are not really affected by this approach, since all we are doing consists of adding and stripping away zeroes from the original values supplied.

BOOK REVIEW

Reviews published in SATN are the opinions of the reviewers, not of Software Arts, Inc. and may not be construed as official positions or endorsements by Software Arts, Inc.

VisiCalc for the IBM Computer, *by Edouard J. Desautels, Wm. C. Brown Company, Dubuque, Iowa, 155 pp., $16.95*

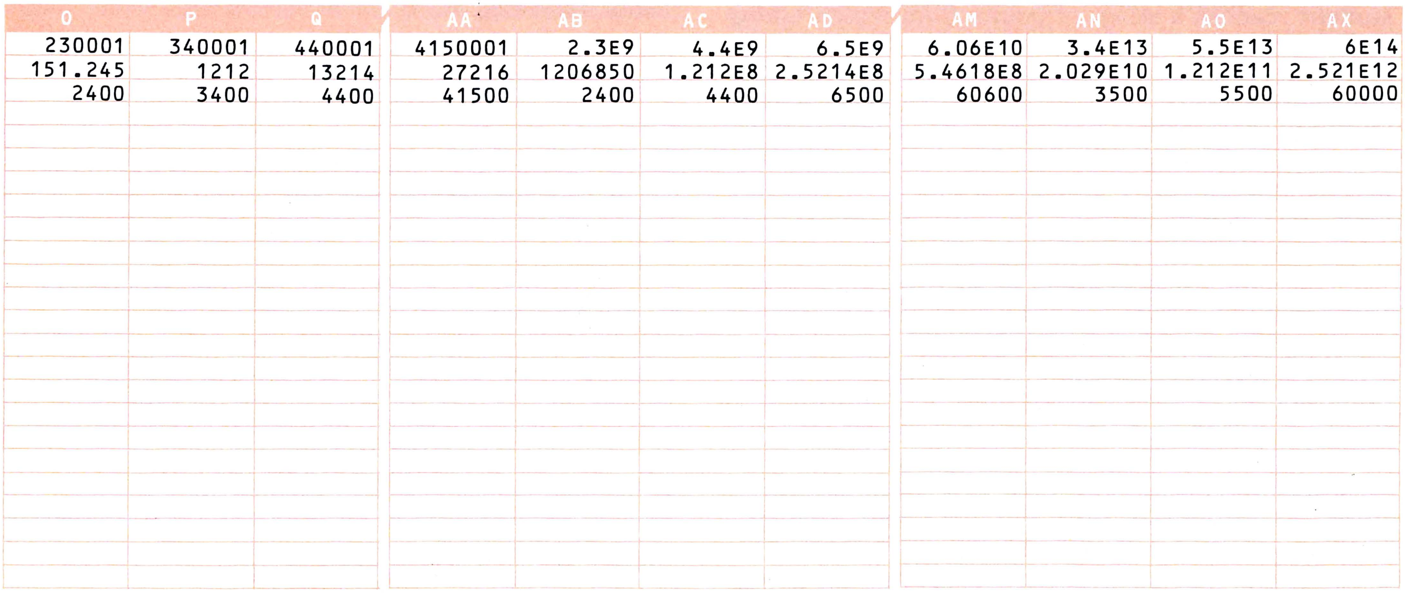
Reviewed by Richard M. Foard

There was a time in the brief history of personal computers when all the books were about the computers themselves, with maybe an appendix or two thrown in about some of the programs that could be run on them. Hardware was king, and it was a foregone conclusion that if one could just lay his hands on the right computer, the right software to do the job would surely follow.

The economics of computing have changed. Today it's hardware vs. software, and nowhere is this more obvious than in the small-computer arena. Informed users now seek the software they need first, the hardware second. In *VisiCalc for the IBM Computer,* Edouard J. Desautels has put the software horse clearly before the hardware cart by quickly disposing of the details of bringing VisiCalc® up on the IBM Personal Computer and then delving deeply and thoroughly into a tutorial explanation of the function and power of VisiCalc.

This is a straightforward book in physical appearance as well as in content. It is spiral-bound and was typeset directly from computer-produced output. Three photographs illustrate the machine and the placement of data diskettes and keys of interest. The rest of the text enjoys no graphics whatsoever, but shows example spreadsheets in typescript tables and spells out key names that don't appear on a typewriter keyboard, like "ENTER."

The text, in direct and simple English, is aimed at the naive, first-time computer user. The first chapter



O	P	Q	AA	AB	AC	AD	AM	AN	AO	AX
230001	340001	440001	4150001	2.3E9	4.4E9	6.5E9	6.06E10	3.4E13	5.5E13	6E14
151.245	1212	13214	27216	1206850	1.212E8	2.5214E8	5.4618E8	2.029E10	1.212E11	2.521E12
2400	3400	4400	41500	2400	4400	6500	60600	3500	5500	60000

but 4, which would handle the amount correctly), then the EFF% OF INCOME will show from 0.30% to 3.30%. Since at very high levels of income the effective rate approaches the marginal rate, such a low figure indicates that you have either exceeded the limits of the model or made a data entry error.

gives a brief pitch for the benefits of using VisiCalc, then explains in detail the parts of the IBM computer and the procedures for inserting and formatting diskettes and for bringing the program up.

Desautels's clarity doesn't suffer from the austere presentation. His pace is gentle, leading the reader through the potentially intimidating first encounter with a computer with reassurances like, "Clearing the screen has no effect on worksheets that have been saved on diskettes."

A few simple formulas introduced in early examples illustrate VisiCalc's automatic recalculation and encourage the reader to explore ways of moving around the spreadsheet. The first chapters set a pattern for the rest of the book by leading the reader through the tedious manual repetition of a formula down a column and then unveiling the beauties of the Replicate command.

The tutorial examples, available on diskette to reluctant typists for $39.95, increase in complexity from a simple royalty calculation to an IRA account benefit analysis, a teacher's gradebook maintenance system, and finally to a set of five case studies including an expense log, stock portfolio evaluation, personal net worth computation, sales projection, and compound interest calculation. Although explained clearly and concisely, these examples are by now VisiCalc classics, and may leave some readers hungry for more innovative applications that illustrate the program's broad flexibility as well as its utility as a replacement for manual spreadsheets.

The book is organized by problem rather than by VisiCalc function. In each example, Desautels introduces another nuance of formatting, replication, or spreadsheet planning and aesthetics. As the author points out in his preface, this organization allows the reader to skip around in the text.

The problems are carefully structured so that the student reaches a series of points from which taking the next step is awkward or confusing using only the facilities presented so far, thus paving the way for introduction of the next topic. The author's thorough work in anticipating VisiCalc's pitfalls for the beginner is apparent. This format deals effectively with potential misunderstandings, but the didactic, repetitive "wouldn't that be difficult . . . not with the /X command!" structure grows a little tiresome.

The telling test to apply to all third-party books about software products is: Do they add significantly to the body of instructional or reference information provided by the manufacturer? It is a particularly demanding test in the case of VisiCalc, because Van Wolverton's IBM product manual ranks among the best in the industry. Clear and well written, it devotes almost half its considerable bulk to a series of tutorial, work-along examples, with the remainder a comprehensive reference guide.

Because Desautels set out to provide an almost purely tutorial text, it's hardly fair to compare his book with Van Wolverton's on the basis of suitability for quick reference. *VisiCalc for the IBM Personal Computer* can, however, be judged against the manufacturer's VisiCalc tutorials. Desautels starts at a more basic level than the VisiCalc manual, assumes less familiarity with computers in his audience, and travels more slowly and carefully while leading the novice. His book is "friendlier."

The book concludes with a refreshingly down-to-earth chapter entitled "When NOT to Use VisiCalc." Here Desautels calls out the characteristics that make some problems better suited for attack by a program written in a general-purpose language than by VisiCalc-assisted ingenuity. His advice will help the first-time user avoid the frustration of tackling problems that just can't be solved on a spreadsheet, even a very versatile electronic one.

Watching an intuitive, analytically inclined person work through his first session with VisiCalc can be an almost awe inspiring experience. After a little orientation and a few missteps, the insights dawn, the intellectual sparks start to fly, and the manual is brushed aside except for occasional reference. The user descends into the depths of the problem and in a matter of hours has joined the ranks of true believers. While this isn't the most methodical or thorough way to learn, it serves many of the people buying computers today quite well. This is not the sort of person who will profit greatly from using Desautels's book.

VisiCalc for the IBM Personal Computer is best put to use by the first-time computer user who still doesn't quite trust the confounded things. It is a competent, clearly written text which has its place in the spectrum of VisiCalc literature. What it lacks in imaginative examples and four-color sizzle, it makes up for in meticulously well-sequenced exposition and a firm grasp of the likely weaknesses of a newcomer to computing. It is a book for anyone who still has anxieties about approaching a computer but who is convinced that VisiCalc is an essential adjunct to his personal or business analytical abilities.

Richard M. Foard is Vice President of software development at Travel Technology, Inc.

BACK ISSUES AVAILABLE

Back issues of SATN are now available. The cost is $6.00 per copy for orders mailed within the USA. If your order is to be mailed outside the USA, add $2.50 for shipping and handling to the full price of the order.

To order back issues, please send your name and address, a list of the specific issues of SATN you want (by Volume and Issue numbers), and a check or money order payable to Software Arts to:

SATN Back Issues
27 Mica Lane
Wellesley, MA 02181

VISICALC AND WORDSTAR

by Craig Stinson

The first article in this series (January/February 1983) discussed VisiCalc®'s /PF command in a general way and specifically considered using the VisiCalc program with Magic Window, a word processing program for the Apple II. We said then that the main reason for using the /PF command is to give you more control over the appearance of printed reports that you generate from VisiCalc models. With /PF you can create an ASCII text file that duplicates the appearance of your model on screen: that file can then be loaded into your favorite word processing program (assuming that the program can read ASCII text files).

Let's look at the mechanics of using a /PF file within a WordStar program on the IBM Personal Computer.

First, note that WordStar is available in two distinct versions for the IBM PC. One runs in the CP/M-86 operating system, the other in PC-DOS, sometimes also called IBM-DOS, MS-DOS, or just DOS, which is how we shall refer to it. Because VisiCalc runs in DOS and creates DOS files, we'll assume your WordStar is of the DOS variety. If that's not the case, you aren't out of luck, but you do need to convert your /PF files to CP/M-86 format before you can work with them. Adaptive Computer Systems (Wheeling, Illinois) markets a utility program called DOSCPM that performs this conversion.

There are three ways to get a /PF file into WordStar. If you are not incorporating the /PF into an existing WordStar document, but wish to work with it directly, you may use WordStar's D or N command (which stand for document and nondocument, respectively). Those commands bring the /PF file directly into WordStar's editor. When working with VisiCalc /PF files you're not likely to need the N, nondocument, mode. It is used primarily to facilitate the editing of programs or other material for which pagination isn't needed. (Pages 6-15 and 6-16 of the WordStar manual summarize the differences between these two editing modes.)

When you want to make your /PF file a part of an existing WordStar document, you must first bring the file into the WordStar editor (using the D command) then move your cursor to the place in the document where you want your VisiCalc model to appear. Once you have brought the /PF file into the WordStar editor you must use the command for inserting a file.

The WordStar command to insert a file from disk into another file already present in the editor is control-KR. The editor will respond by prompting you for the name of the file you wish to insert. When you supply it, remember that the VisiCalc program has attached the extension .PRF to the name you gave your /PF file while you were still in VisiCalc. When your /PF file appears, you'll find a line of carets and at-signs (@). The VisiCalc program used them to mark the end of the file, and WordStar does not read them that way. You can get rid of them by putting your cursor anywhere on that line and typing control-Y.

You can load as many /PF files into a single document as you wish. One of WordStar's strengths is you can create and edit very large documents; file size is limited by the space available on disk, not by the amount of random access memory in your machine. With WordStar, you can link a number of /PF files to produce a printed report that might have been too large to achieve by overlaying files within the VisiCalc program.

In WordStar, control-K is a generic prefix for commands that affect a block of text. Control-KR allows you to read a block of text. For the R, by the way, you may substitute control-R, if you prefer to leave your finger on the control key. In all two-character WordStar commands, the second character may be either an ordinary letter (capital or lower case) or its control equivalent.

The opposite of control-KR, control-KW, is also useful for manipulating /PF files. Control-KW allows you to save a portion of a WordStar document as a separate file. Control-KW is handy in cases where you want to juxtapose a portion of one VisiCalc model with all or a portion of another.

To do so, you first have to mark the beginning and end of the text you wish to extract. Move the cursor to the beginning of the area you want to save, hit control-KB, and a marker (<B>) will appear. Then move to the end of the area to be saved, hit control-KK, and another marker (<K>) will appear. The portion of the text that you have defined by placement of the markers will change from high intensity to low intensity on your video display. Now issue the control-KW command, answer the prompt for a file name, and off it goes. The defined area is copied to disk by this procedure. If you want to leave the text in the parent document, but get rid of the block markers after you've copied it to disk, use the command control-KH. If you want to remove it from the parent document, follow control-KW with control-KY, for delete.

To incorporate VisiCalc-generated data into text reports, WordStar's direct page break and conditional page break functions may be used. With the direct page break command you can force WordStar to begin printing your VisiCalc data at the top of a page, regardless of where that data may be located within the parent document. Simply put the characters ".pa" all the way to the left on a line by themselves immediately above your VisiCalc data. Your screen will display a row of hyphens, indicating a page break, directly below the .pa command.

The conditional page break command works a little differently. The code is ".cp" followed by a number. There should be nothing else on the same line, just the code and number. WordStar will count the number of lines remaining on the current page (below the position of the .cp command). If that number is lower than the number you specify, the program issues a page break. Even if your model is small and clearly fits on a page, a conditional paging command is a good idea. It allows you to add text later on without worrying about whether the addition will force a page break within your model.

Everything thus far assumes that you're planning to print your /PF file without editing it, which is probably not the case. So now let's look at some specifics about editing tabular material in WordStar.

When you bring your /PF file into the WordStar editor, its appearance is not affected by the margin settings in effect or by the current line spacing value. WordStar will not try to constrain your model to fit the existing margin settings, unless you happen to hit control-B.

If you issue that command, WordStar will "reform" whatever line your cursor is on to fit within the designated margins. If your /PF file is wider than the margin settings and you want to avoid accidental reformation, there are a couple of things you can do (besides staying away from control-B). You can type control-OX, which suspends left and right margin settings, just like the margin release key on a typewriter. Control-OX is a toggle; type it a second time to restore the margin settings.

You can also put your cursor anywhere on the widest line of the model and type control-OF, to reset the right and left margins to match the width of the line on which the cursor is positioned. Control-OF also clears any tab settings and sets new tab stops in any column where it finds an exclamation point or a pound sign (#). If you're going to do much editing of your VisiCalc data, you'll probably want to set new tabs to match your VisiCalc columns anyway, so you won't mind clearing the existing stops first.

WordStar offers two types of tab settings: normal and decimal. To set a normal tab stop, type control-OI, followed by either a number designating the column in which you want the tab to be set, or by the escape key, which puts a tab setting at the position where your cursor happens to be located. The procedure to establish decimal tab stops (more about them in a moment) is the same, except that you hit the pound sign (#) before you type the column number or hit the escape key. If you want to set up tabs to match the format of your VisiCalc data, you'll find control-F useful; that command hops from column to column in your model, and at each stop along the way you can enter a control-OIescape or control-OI#escape.

Decimal tab stops are useful for keeping numbers correctly aligned while you manipulate them. For example, suppose you have a column of numbers in which you need to insert commas. Put your cursor over the decimal point in one of these numbers (or directly to the right of the ones-place number, if there aren't any decimal points) and hit control-OI#escape to set the decimal tab. Now you can tab to the decimal tab stop (you should see the word DECIMAL at the top of your screen when you do), backspace three characters, go into insert mode, and type your comma. The number will stay properly aligned.

Two caveats apply to this procedure: you must arrive at the decimal tab stop by means of the tab key in order for the decimal feature to be active, and while tabbing around on your screen you must NOT be in insert mode. If you hit the tab key while you're in insert mode, you simply insert a lot of space and throw your columns out of line.

And speaking of getting columns out of line, if you need to erase figures, the best way to do it is to strike over them with blank spaces. If you use the normal character or word delete functions (control-G or control-T), you not only delete the numbers where your cursor lies, you also pull all the numbers to the right of your cursor position a corresponding number of positions leftward.

Function-5 and Function-6 produce local printer codes, effecting underscoring and boldface, respectively. Both act as toggles; the first Function-6, for example, turns boldfacing on, and the second turns it off.

If your printer has a condensed mode, you turn that on and off by means of control-PA and control-PN, respectively. But first you have to provide the WordStar program with the setup routines appropriate for your printer. You'll need to consult the WordStar Installation Manual (available on request from MicroPro International) or your dealer for help in this matter.

Control-PA and control-PN allow you to print a section of your file in condensed mode and then return to normal mode. That's extremely handy if you have a wide model that you want to incorporate into a document of normal width.

Last but not least, WordStar has a block-column feature that allows you to perform block operations on selected columns of data with a /PF file. You can, for example, transpose columns, copy columns, or even consolidate columns of information from one model with columns from another. To turn the block-column feature on, type control-KN. This is another toggle code; the second time you type it, you turn the feature off.

The difference between the normal block mode and the block-column mode is that an area defined in the former includes all text lying to the right of the beginning marker on the line where it's placed, to the left of the ending marker on the line where it's placed, and all intervening lines in their entirety. In block-column mode, the beginning and end markers define the corners of a rectangle.

In either mode, once you've defined an area you have the options of moving it (control-KV), copying it (control-KC), and deleting it (control-KY). Control-KH removes existing block markers.

The control-KW command is not available in block-column mode. Nevertheless you can write selected columns from a VisiCalc model to disk, if you enter block-column mode, define the area you wish to save, and copy it to a position at the end of the file. Then hit control-KN to return to the normal block mode, define the copied columns now located at the end of your file, send that block to disk with control-KW, and erase the block from your parent file with control-KY.

Craig Stinson is the editor of Softalk for the IBM PC.

RELATED PRODUCTS CHECKLIST

The information contained in this column has been garnered from a variety of sources, including dealers, press releases, manufacturers, and advertising.

No one at Software Arts, Inc. has reviewed or examined any of these products to determine whether they perform as claimed or as listed below. An inclusion in this listing does not constitute an endorsement by Software Arts, Inc. of that product. We also wish to point out that if any of the products listed alter, patch, or copy the VisiCalc program, they could be in violation of your user licensing agreement and/or the copyright.

Since we have not reviewed any of the products, we cannot provide you with any more information than is contained in the list. If you want to learn more or you wish to purchase any of the products listed, contact your dealer first. Although we have identified distributors in some listings, many distributors do not accept orders from individuals for single copies of software or pieces of hardware.

SOFTWARE

Only software that claims to complement the VisiCalc program will be listed.

Agricultural Templates

FARM AIDS

- Runs on Apple][

Templates for best crop, finishing hogs, feeder pig production, beef feeders, dairy production, cash flow, inventory, net worth.

Distributed by SDSI, Box 8278, Madison, WI 53708.

Suggested retail price: $60.00.

Real Estate Templates

Real Estate Models for the Eighties

- Runs on Apple][, Apple][+, Apple ///, IBM PC

Real estate analysis using "creative financing" techniques.

Distributed by Commercial Software Systems, 7689 W. Frost Dr., Littleton, CO 80123.

Suggested retail price: $65.00.

Financial Templates

Investment Tax Analyst

- Runs on IBM PC

Investment analysis combined with a tax calculation and projection system. Estimates Federal income taxes both before and after any of a series of investments is made.

Distributed by Advanced Investment Strategies, Inc., 1007 Massachusetts Ave., N.E., Washington, DC 20002.

Suggested retail price: $125.00.

Personal Financial Statements

- Runs on Apple][, Apple ///, and IBM PC

For financial planners, models client's present financial strategy, showing investments and expenses, and projects a proposed strategy. Many factors can be adjusted to increase net worth before or after taxes.

Distributed by United Financial Modelers, 26 Main St., Concord, MA 01742.

Suggested retail price: $725.00.

Graphics

Chartdif and Plotdif

- Run on IBM PC
- Support DIF format

Utility programs to interpret VisiCalc data in charts and plots. Programs are sold together.

Distributed by The Consultant, 414 Can-Data, Mt. Prospect, IL 60056.

Suggested retail price: $75.00.

Word Processors

Electric Pencil 2.0

- Runs on TRS-80 Model I and TRS-80 Model III

Upgrade to Electric Pencil 1.0: expanded print and system menus, selective search and replace, selectable cursor speed and the ability to save print formats.

Distributed by IJG, Inc., 1260 West Foothill Blvd., Upland, CA 91786.

Suggested retail price: $89.95.

Wordcraft80

- Runs on Commodore 8032

Word processor, supports underlining, superscripts, subscripts, bold, overprinting, and bidirectional printing. Text appears on screen very much as it will appear when printed. VisiCalc files can be accessed and manipulated.

Distributed by Commodore Business Machines, 681 Moore Road, King of Prussia, PA 19406.

Suggested retail price: $300.00.

Consolidators

The Context Connector

- Runs on Apple][

Program to consolidate any number of VisiCalc models into summary models; reports, databases and spreadsheets from other computers can be included in your VisiCalc models.

Distributed by Context Management Systems, 632 Via Del Monte, Palos Verdes, CA 90274.

Suggested retail price: $180.00.

Accounting and Bookkeeping

Maxi Cras

- Runs on TRS-80 Model I and TRS-80 Model III

Accounting program to print checks, balance accounts, reconcile bank statement and support as many as 223 income and expense accounts. Interfaces with VisiCalc.

Distributed by Adventure International, P.O. Box 3435, Longwood, FL 32750.

Suggested retail price: $99.95.

To submit information on products (software or hardware) that support either the VisiCalc program or the DIF format for data exchange, send us the name of the product; what it does; who developed it; who markets and distributes it; the suggested retail price; if it is software, what machine(s) it runs on and whether it supports the DIF format. Send the information to:

SATN Products Column
27 Mica Lane
Wellesley, MA 02181

VISICALC ADVANCED VERSION

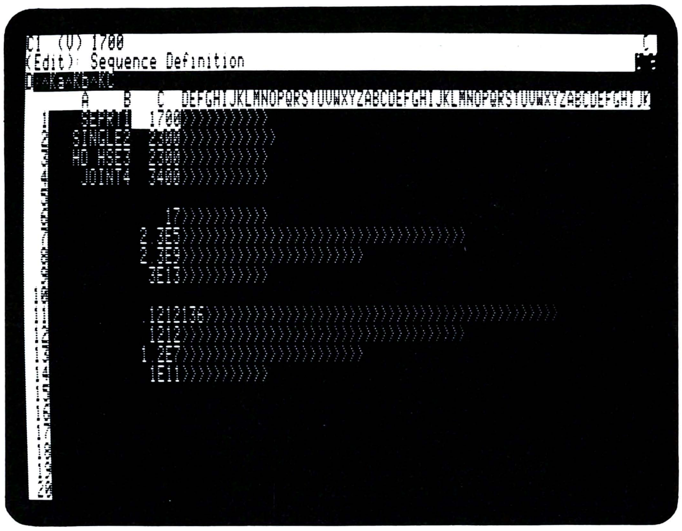
Although *Turning the Tables on Taxes* does not use any features of the VisiCalc® program beyond those available in all versions of the program, we used the Advanced Version of the program to develop the sheets used in the article. As we put the sheets together, of course, we explored some of the ways we could use some of the Advanced Version's unique features.

Viewing the Entire Worksheet

Creating the large @LOOKUP table used in the article on taxes called for a series of replications to turn twelve lines into three. Each succeeding replication dealt with a longer source range and meant that it was necessary to scroll the beginning of the range off the screen while scrolling to the end.

If you are using the VisiCalc Advanced Version program, however, you can set the column width to 1 character, which will allow you to view all 63 columns of the worksheet. As you can see from Figure 4, only a few of the actual values will be displayed, with the rest appearing as "greater than" signs. Although you will not be able to read the values, since you now have every column in sight, you can line up the different ranges visually and execute the entire series of replications by positioning the cursor appropriately, rather than by entering the specific coordinates.

When we developed the sheet for *Turning the Tables on Taxes,* we used the VisiCalc Advanced Version Keystroke Memory feature to capture the keystrokes needed to perform each of the replications the first time that we executed them. After performing the first series, we chained the three sequences together to create a single sequence that made carrying out the replications quite easy. In Figure 4 we show the chained sequences on the edit line and discuss one of the sequences in detail in the caption to that figure.



```
C1  (V) 1700
(Edit): Sequence Definition
D:^Ka^Kb^Kc
      A    B    C  DEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJ
     SEPRT1  1700>>>>>>>>>>>
    SINGLE2  2300>>>>>>>>>>>
    HD HSE3  2300>>>>>>>>>>>
     JOINT4  3400>>>>>>>>>>>

               17>>>>>>>>>>>
            2.3E5>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
            2.3E9>>>>>>>>>>>>>>>>>>>>>>
             3E13>>>>>>>>>>>

         .1212136>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
            .1212>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
            1.2E7>>>>>>>>>>>>>>>>>>>>>>
             1E11>>>>>>>>>>>
```

FIGURE 4 *Notice that columns A and C are five characters wide, not one. Only a portion of the replications have been completed. The next will copy row 7 into row 6.*

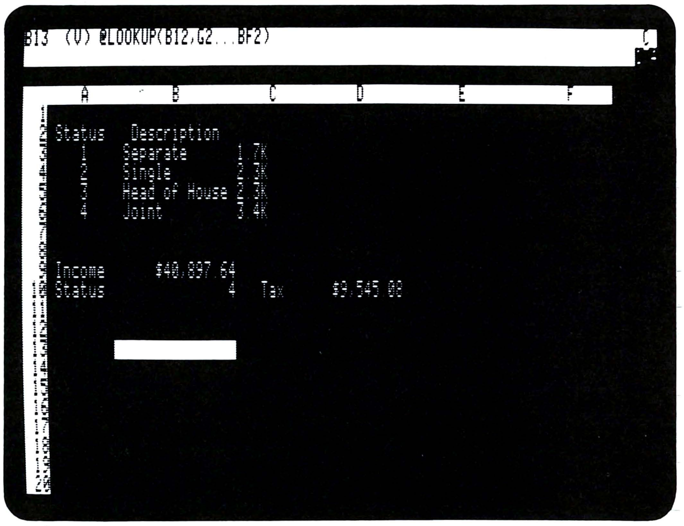
To copy the lowermost row into the next higher one in any of the three tables, we created the keystroke sequence ^KA by invoking ^K= and then using the cursor. The result was the sequence definition:

```
A:/R^>^>^>^>^>^>^>^>^>^>^>^R^U
^>^>^>^>^>^>^>^>^>^>^>^>^R^U
```

Cryptic as it appears, the sequence is relatively easy to read. The /R starts the replicate command. Each ^> moves the cursor one column to the right (^< moves it to the left). The ^R is read as a carriage return. The whole sequence from /R through the first ^R defines the source range. If you were to enter that sequence from the keyboard, you would define a source range that is eleven columns wide. The target range would begin one row above (the first ^U), and twelve columns to the right of the start of the source range (remember that the cursor returns to the beginning of the source range after it has been defined). The second ^R defines the beginning of the target range, and since we are replicating a range into a range in the same direction, we need only enter the beginning of the target range. Since the values being replicated do not consist of expressions, there are no value references to include in the sequence. The final ^U positions the cursor on the row above the one that has just been replicated in preparation for the definition of the next sequence. *Photograph by Dana Sigall*

Hiding Your Work

Rather than use /F∗ as we did in the tax article to hide intermediate steps in the tax calculation, we can actually hide intermediate calculations with the Hide Attribute. As you can see from Figure 5, the only values that are actually displayed are taxable income, status, and the tax (the calculation of which is in a different position on the Advanced Version sheet). In order to hide the tables and intermediate calculation, we entered /AHN (Attribute, Hide, No) at A2 and then replicated it with /R(A) (for replicating attributes only) into the cells that are visible in the figure. Then we entered /GAHY (Global, Attribute, Hide, Yes) to hide all the remaining entries.



```
B13  (V) @LOOKUP(B12,G2...BF2)

      A          B          C         D          E          F
  Status   Description
     1     Separate     1.7K
     2     Single       2.3K
     3     Head of House 2.3K
     4     Joint        3.4K


  Income      $40,897.64
  Status               4  Tax     $9,545.08
```

FIGURE 5 *Notice that although nothing appears on the screen where the cursor is resting, the contents line of the status panel shows an expression (with slightly different range coordinates from the ones used in the main article).* *Photograph by Dana Sigall*